Tightening: Curvature-Limiting Morphological Simplification

Jason Williams Jarek Rossignac

GVU Center, College of Computing, Georgia Institute of Technology

{jasonw, jarek}@cc.gatech.edu

ABSTRACT

Given a planar set S of arbitrary topology and a radius r, we show how to construct an r-tightening of S, which is a set whose boundary has a radius of curvature everywhere greater than or equal to r and which only differs from S in a morphologically-defined tolerance zone we call the mortar. The mortar consists of the thin or highly curved parts of S, such as corners, gaps, and small connected components, while the boundary of a tightening consists of minimum-length loops through the mortar. Tightenings are defined independently of shape representation, and it may be possible to find them using a variety of algorithms. We describe how to approximately compute tightenings for sets represented as binary images using constrained, level-set curvature flow.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling – *geometric algorithms, languages, and systems.*

Keywords

mathematical morphology; topology; simplification

1. INTRODUCTION

Although the set theoretic formulation of the proposed approach is independent of dimension, we restrict our discussion and implementation to full-dimensional subsets of the plane. Our input is a set S that is the closure of its interior. Let ∂S denote the boundary of S and $\bar{S}$ its complement.

The complexity of S may be measured in a variety of ways. Tightening decreases several of these measures. It increases smoothness (the minimum radius of curvature of ∂S.) Furthermore, it tends to reduce topological complexity (the number of components of S and $\bar{S}$,) increase the least feature size (the minimum distance between ∂S and its medial axis,) and decrease perimeter (the length of ∂S.) Tightening does not modify the r-regular portions of ∂S, which lie further than r from the medial axis of ∂S. Instead, it confines its effect to the mortar, which is typically a small subset of all the points within a distance r of ∂S.

We begin by defining r-tightenings and then describe how to compute an r-tightening for an input set represented as a binary image. We conclude with a discussion of tightenings in relation to prior art.

2. DEFINITIONS

2.1 R-Smoothness

A G^1-continuous curve is *r-smooth* iff its radius of curvature is greater than or equal to zero everywhere it is defined. A set is r-smooth iff its boundary is r-smooth.

2.2 Growing and Shrinking

The mortar is defined in terms of the morphological operations of rounding and filleting, which can in turn be defined in terms of growing and shrinking. S *grown* by r, denoted $S\uparrow_r$, is the set of all points whose distance from S is less than r (Figure 1 (a)). It does not include its boundary, so it is topologically open. S *shrunk* by r, denoted $S\downarrow_r$, is the set of all points whose distance from $\bar{S}$ is greater than or equal to r (Figure 1 (b)). It includes its boundary, so it is closed. Shrinking eliminates small or thin pieces of S, while growing fills in gaps and cracks. Both operations can change the topology of S by, for example, changing the number of components of S or $\bar{S}$.

We highlight properties of growing and shrinking by providing alternative definitions equivalent to those given above. If we let B_r denote an open disk of radius r, then $S\uparrow_r$ is the union of all B_r whose centers lie in S, while $S\downarrow_r$ is the complement of the union of all B_r whose centers lie in $\bar{S}$. This formulation shows that growing and shrinking are dual: $\overline{S\uparrow_r} = \bar{S}\downarrow_r$, so that, for instance, the same result may be achieved by growing a set as by shrinking its complement. It also implies that the convex portions of $\partial\left(S\uparrow_r\right)$ are r-smooth, while symmetrically, the concave portions of $\partial\left(S\downarrow_r\right)$ are r-smooth.

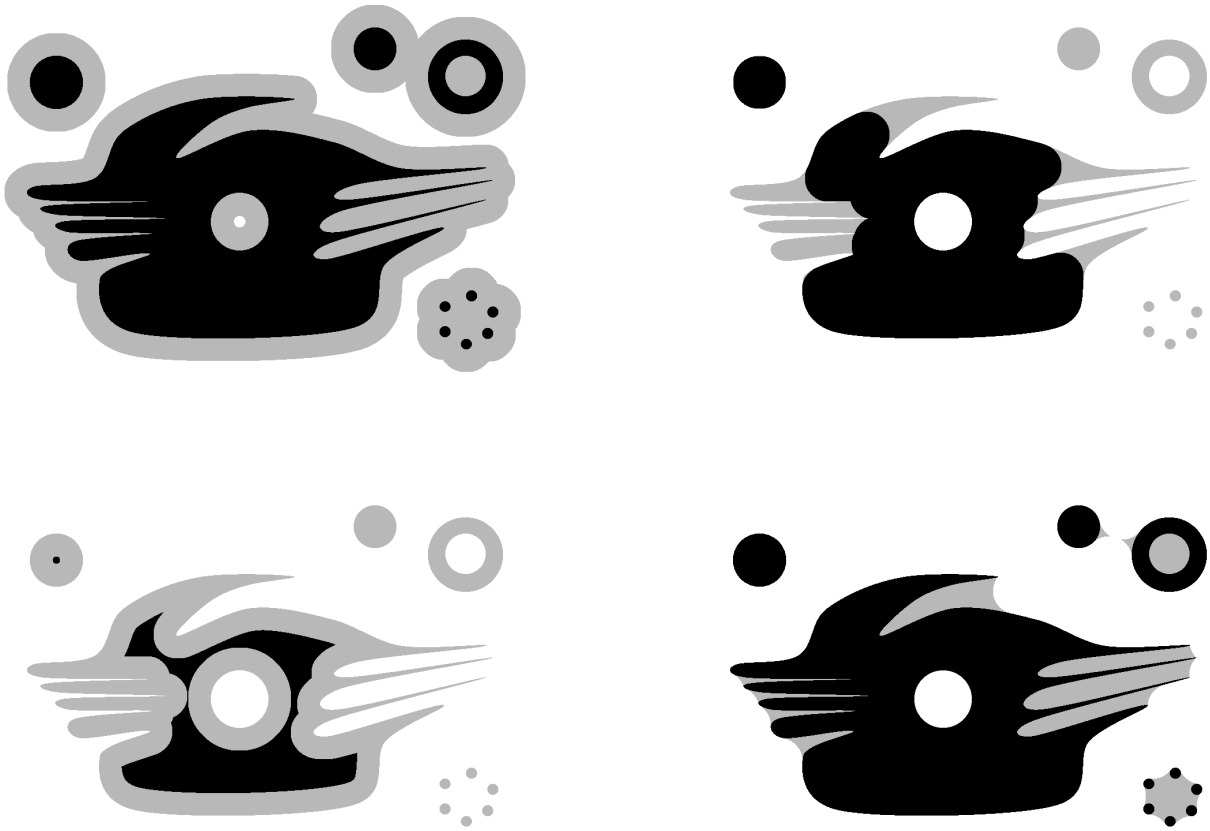


Figure 1 (a) The input set S is shown in black, and the material $S\uparrow_r$ adds to S is shown in gray. (b) $S\downarrow_r$ is shown in black, and the material it removes from S is shown in gray.

$\partial\left(S\uparrow_r\right)$ are r-smooth, while symmetrically, the concave portions of $\partial\left(S\downarrow_r\right)$ are r-smooth.

2.3 Rounding and Filleting

S *rounded* by r, denoted $R_r\left(S\right)$, is $S\downarrow_r\uparrow_r$ (Figure 2 (a)). $R_r\left(S\right)$ is the union of all B_r that lie in S. This can be seen by considering that $S\uparrow_r$ can also be described as the centers of all B_r that lie in S. When we grow the centers of the disks, we obtain the disks themselves. It follows that, like $S\uparrow_r$, $R_r\left(S\right)$ is open and the convex portions of its boundary are r-smooth. We can also conclude that $R_r\left(S\right)\subseteq S$. The set $S-R_r\left(S\right)$ contains the points that cannot be included in any B_r disjoint from ∂S. It includes convex corners and the thin constrictions removed by shrinking.

Figure 2 (a) $R_r\left(S\right)$ is shown in black and the material it removes from S is shown in gray. (b) S is shown in black and the material $F_r\left(S\right)$ adds to S is shown in gray.

The dual of rounding is filleting. S *filleted* by r, denoted $F_r\left(S\right)$, is $S\uparrow_r\downarrow_r$ (Figure 2 (b)). $F_r\left(S\right)$ is the complement of the union of all B_r that lie in $\overline{S}$. Like $S\downarrow_r$, $F_r\left(S\right)$ is closed and the concave portions of its boundary are r-smooth. $F_r\left(S\right)\supseteq S$, and $F_r\left(S\right)-S$ contains the portions of $\overline{S}$ in the concave corners of S as well as the gaps of S filled by growing.

2.4 Mortar, Core, and Anticore

The *mortar* $M_r\left(S\right)$ is $F_r\left(S\right)-R_r\left(S\right)$. The mortar is also closed and the concave portions of its boundary are r-smooth. Being the union of $F_r\left(S\right)-S$ and $S-R_r\left(S\right)$, it contains all of the details associated with S, including sharp convexities and concavities as well as the thin parts of both S and $\overline{S}$. Away from these details, the mortar is identical to the

boundary of S. Because it can contain these lower- dimensional portions, the mortar may not be equal to the closure of its interior.

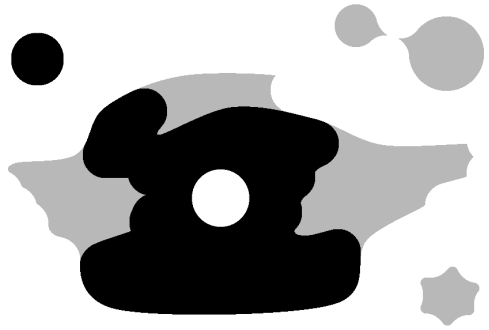


Figure 3. The core (black,) mortar (gray,) and anticore (white) for the set S shown in Figure 1.

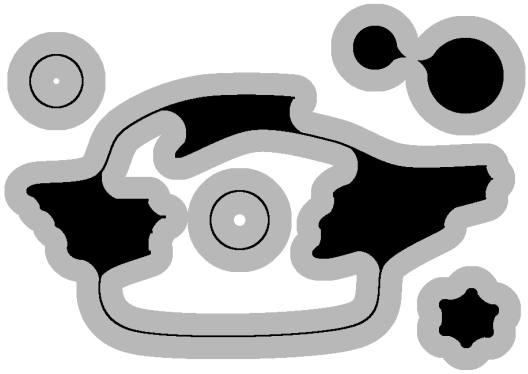


Figure 4. $M_r(S)$ (black) and $Z_r(S)$ (black and gray) for the set S shown in Figure 1.

We use the mortar as a tolerance zone, and to emphasize that they are not affected by tightening we refer to $R_r(S)$ as the *core* and $\overline{F_r(S)}$ as the *anticore* of S. Together, the core, mortar, and anticore decompose the plane into three mutually disjoint sets, where the mortar is closed and the core and anticore are open (Figure 3).

We can compare the mortar to the tolerance zone $Z_r(S)$ consisting of all points within a distance r of ∂S (Figure 4). $Z_r(S)$ is $(\partial S)\uparrow_r$, while the mortar can be expressed as $F_r(\partial S)$, which is $(\partial S)\uparrow_r\downarrow_r$. Consequently, $M_r(S) = Z_r(S)\downarrow_r$, so that the mortar is restricted to points of $Z_r(S)$ that are at a distance of r or more from $\overline{Z_r(S)}$.

2.5 Tight Loops

Tightenings are bounded by minimum-length loops through the mortar. We say that these boundary loops are tight with respect to the mortar, where we define a simple closed curve C lying in a closed set X as *tight* with respect to X iff it locally minimizes length, so that there exists an ε such that for all $\delta \le \varepsilon$, $d(C(t), C(t+\delta)) = \delta$, where $d(A,B)$ is the length of the shortest path connecting A and B in X and C is parameterized by arclength. C is tight iff it is the shortest curve that may be obtained by a continuous deformation of C that keeps it within X.

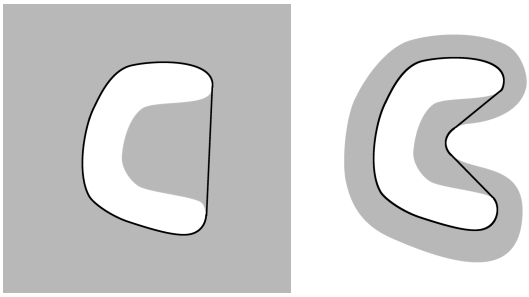


Figure 5. The set X is shown in gray while the curve C that is tight with respect to X is shown in black. Left: X is the plane minus a hole. Right: X is a crescent-shaped annulus.

Our first observation is that if C is tight with respect to X, it must surround one or more components of $\overline{X}$. Consider, for instance, the case where X is the plane minus a hole (Figure 5, left). Then the only curve tight with respect to X is the convex hull of the hole. Next consider the case where X is a crescent-shaped annulus (Figure 5, right). There is still only one tight curve C. It contains the portion of the convex hull of the hole that lies along the boundary of the hole, but it is now pushed inwards by a concave piece of the outer loop of X. The parts of C that lie along ∂X are smoothly connected by straight line segments that lie in the interior of X. This situation holds in general: tight curves through a set consist of concave portions of its boundary connected by tangent line segments.

2.6 R-tightenings

An *r-tightening* of S is a set T such that $R_r(S) \subseteq T \subseteq F_r(S)$ and the boundary of T consists of disjoint loops that are tight with respect to $M_r(S)$. Because the concave portions of the boundary of the mortar are r-smooth, the conclusion of the previous section yields the following theorem:

Theorem: *The boundary of an r-tightening is r-smooth.*

If the core and anticore of S each consist of a single connected component, the tightening is unique, and its boundary is the

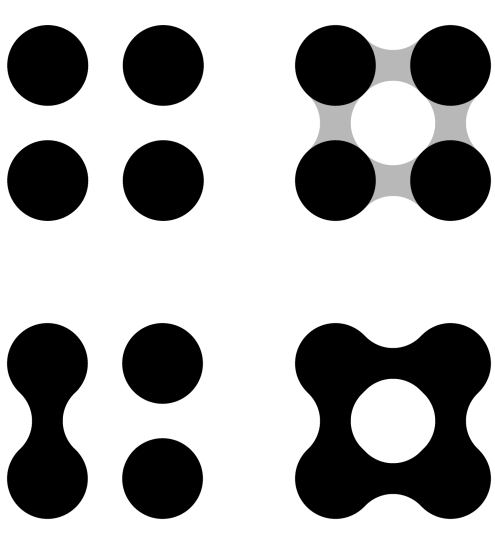


Figure 6. Top left: The input set S. Top right: The core (black,) mortar (gray,) and anticore (white) of S. Bottom left: A tightening of S with three components. Bottom right: A tightening of S with a hole.

shortest loop around $R_r(S)$ lying in $F_r(S)$. When the core and anticore have more complex topologies, there may be several different tightenings, which may have holes and multiple connected components (Figure 6).

3. IMPLEMENTATION

In this section we present an algorithm for computing an r-tightening for an input set represented as a binary image. Our strategy for computing an r-tightening of S is to deform the bounding loops of S in a way that reduces their length while constraining them to lie in the mortar. If we preserve the loops' topology as we tighten them, some loops may eventually overlap themselves, or distinct loops can overlap (Figure 7). Then, although the loops individually are smooth, the set they bound can have cusps. We can, however, arrive at a set with a smooth boundary if we let the loops split, merge, or disappear as they evolve. We opt to perform the evolution in a level set framework, which eliminates the need to explicitly handle these topology changes.

In our implementation the evolving set S' corresponds to a function $\Phi(t,x,y)$ represented at a given time by floating point values on a pixel grid. We consider pixels with negative values to be in S' while pixels with positive values are out. We move the boundary of S' by increasing or decreasing values that are close to the zero level set.

We use curvature flow for the boundary motion. In the continuous case, curvature flow moves a point on a curve along its normal at a speed proportional to the curvature at that point. This dampens oscillations by pushing convex boundary in and concave boundary out. In the absence of constraints, it eventually deforms a curve to a circle which collapses to a point [11]. Curvature flow is well-suited to our

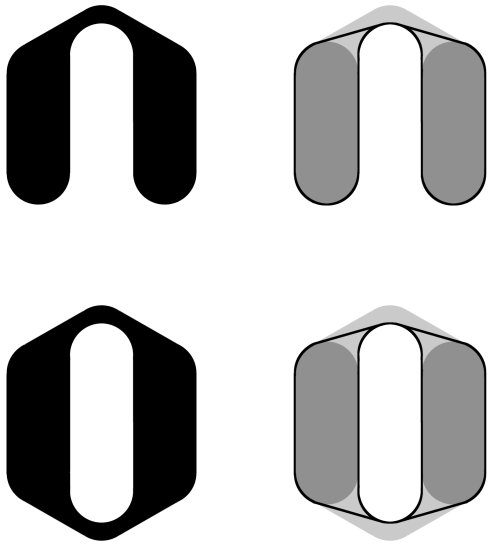


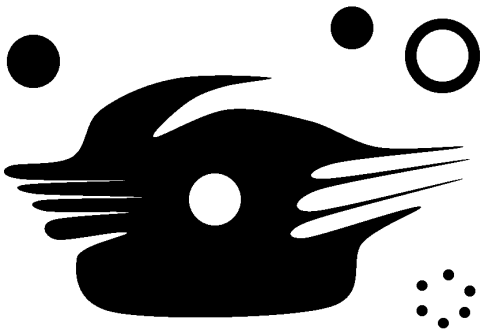
Figure 7. Top left: The input set S. Top right: The core of S is shown in dark gray, the mortar is light gray, and the loop obtained by tightening ∂S while preserving its topology is shown in black. The set bounded by the loop has two cusps. Bottom: A similar example where two loops overlap

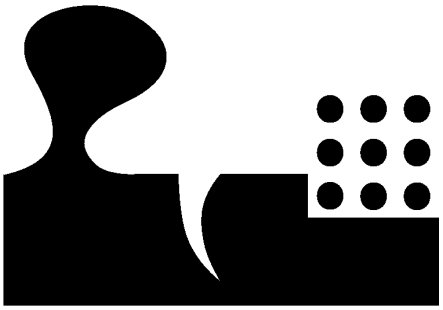
purposes because it rapidly reduces curve length. It is in fact often referred to as "curve shortening flow."

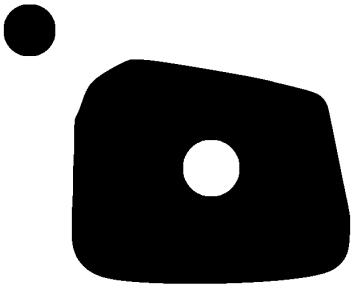
Before starting the flow, we compute the core and the anticore using a combination of growing and shrinking operations. To grow or shrink a set, we threshold its signed distance transform, which we obtain using Danielsson's approximate vector propagation algorithm[7]. This algorithm runs in time linear in the number of pixels. It computes a vector offset from each pixel in the input set to the closest pixel in its complement, and from each pixel in the complement to the closest pixel in the set. It propagates these vectors outward from boundary pixels by sweeping left and right across each scanline, visiting scanlines first from top to bottom and then from bottom to top. It determines the vector at the current pixel in the sweep from the vectors at its neighbors. Although this procedure is not exact, the maximum error is only 0.29 pixels, and it is both faster and easier to implement than several of the alternatives.

Our implementation of curvature flow follows the description in [22]. To start the flow, we initialize Φ to be the signed distance to the boundary of the core of S using vector propagation. At each iteration, we compute $\Phi_t = -F|\nabla\Phi|$ at each pixel, where F is the velocity of the level set. The velocity is zero in the core and anticore. Consequently they remain respectively inside and outside the evolving set. F is equal to the curvature in the mortar, where the curvature at a point is equal to the divergence of the unit normal to the level set passing through that point. Because the normal to the level set is the gradient of the level set function, the formula for the curvature κ is:

$$\kappa = \nabla \cdot \frac{\nabla\Phi}{|\nabla\Phi|} = \frac{\Phi_{xx}\Phi_y^2 - 2\Phi_x\Phi_y\Phi_{xy} + \Phi_{yy}\Phi_x^2}{\left(\Phi_x^2 + \Phi_y^2\right)^{3/2}}$$







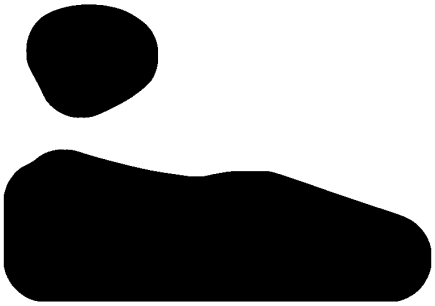


Figure 8 Top: The input set. The image is 1102 x 832 pixels. Bottom: The tightening of the input computed with a radius of 40 pixels.

The partial derivatives are computed using finite differences. Once we have the values for Φ_t, we update the level set function values using

$$\Phi\left(t + \Delta t, x, y\right) = \Phi\left(t, x, y\right) + \Delta t \cdot \Phi_t\left(t, x, y\right)$$

We use a value for Δt that is inversely proportional to the maximum value of F at time t, so that the level set crosses at most one pixel each iteration.

We accelerate this computation using a narrow band technique [22]. We traverse the volume and build a list of pixels adjacent to pixels of the opposite sign. We then use breadth-first search to expand this list to include all pixels within a chessboard distance of three pixels from a sign change. We confine updates to Φ to pixels in this band as long as the zero level set remains within the band. When it leaves the band we traverse the volume again to construct a new band. Overall this achieves an order of magnitude speedup.

Curvature flow converges slowly where the radius of curvature spans several pixels. To accelerate convergence, we devised a new technique where we downsample the image representation of the core by a factor of two until r slightly exceeds the size of a pixel. We perform the flow at this coarse resolution, then iteratively upsample by a factor of two and re-perform the flow. In principle there are cases where the mortar can be arbitrarily thick and the zero level set must consequently cross a large number of pixels at the coarsest level of resolution. The examples we have tested, however, have required at most 100 iterations at each level of resolution.

Figure 9 Top: The input set. The image is 1050 x 825 pixels Bottom: The tightening of the input computed with a radius of 75 pixels.

Figure 8 shows the tightening of the set from Figure 1 computed using this technique with a radius of 40 pixels. It took five minutes to compute the morphological operations for the 1102 x 832 pixel input image and another seven minutes to perform the curvature flow. Figure 9 (bottom left) shows another input image and Figure 9 (bottom right) shows its tightening with a radius of 75 pixels. The image size for the second example is 1050 x 825 pixels, so the execution time is similar to the first example.

Like continuous curvature flow, constrained level set curvature flow decreases the length of the zero level set. Because this length cannot decrease without bound, it converges to a fixed value. This happens if and only if the loops in the zero-level set are tight; curvature flow will not affect a tight loop, and it will shorten a loop that is not tight. We can conclude that level set curvature

flow converges to an r-tightening, which entails that an r-tightening exists.

4. DISCUSSION IN THE CONTEXT OF PRIOR ART

4.1 Mathematical Morphology

Growing, shrinking, rounding, and filleting are examples of operations developed in the field of mathematical morphology. There is a large literature on mathematical morphology; classic texts include [14] and [21]. In this literature, growing and shrinking are referred to as the Minkowski sum and difference with an open disk, or as dilation and erosion with a disk. Rounding and filleting are referred to as opening and closing with a disk. The open disk B_r is called the structuring element.

4.2 Morphological Simplification

Tightening is closely related to combinations of rounding and filleting such as $R_r\left(F_r\left(S\right)\right)$ and $F_r\left(R_r\left(S\right)\right)$ [20]. Like tightening, these filters only change the shape in the mortar. They also tend to produce shapes that are r-regular [2], where a set S is r-regular if $R_r\left(S\right) = F_r\left(S\right)$, so that the mortar is empty and both S and $\bar{S}$ are a union of B_r. Regularity implies smoothness, so $R_r \circ F_r$ and $F_r \circ R_r$ also tend to produce shapes that are r-smooth. However, $R_r\left(F_r\left(S\right)\right)$ and $F_r\left(R_r\left(S\right)\right)$ are not guaranteed to be regular or smooth. They may contain irregularities such as cusps and constrictions. In [29] we proved that for some S there is no r-regular shape that only differs from S in the mortar. However, with tightening we can construct a shape that is r-smooth. This shape has no cusps, although it may still contain constrictions.

In [29], we also pointed out that $R_r \circ F_r$ and $F_r \circ R_r$ are not self-dual, where a filter Ψ is self-dual iff $\overline{\Psi\left(S\right)} = \Psi\left(\bar{S}\right)$. This lack of self-duality means that they do not treat positive and negative space symmetrically, and in particular $R_r \circ F_r$ is biased to add material in regions containing details, while $F_r \circ R_r$ is biased to remove it. This led us to introduce the self-dual Mason filter, which replaces each connected component of the mortar's interior with the result of applying $R_r \circ F_r$ or $F_r \circ R_r$, depending on which yields a smaller symmetric difference with the input set. In this paper we have used the term "tightening" to refer to a procedure that computes r-tightenings, but we have not defined a specific tightening function, so we cannot assert that tightening is or is not self-dual. It is possible to define a self-dual tightening function. For instance, the function that maps S to the result of performing constrained level-set curvature flow with S as the initial condition is self-dual. If we perform the flow with $\bar{S}$ the core and the anticore change roles but the mortar and the front propagation within the mortar remain the same.

We can consider this filter to be an improvement on prior art in morphological simplification because we regard self-duality and smoothness as desirable properties. However, the filter may fail to eliminate constrictions that would be removed by a combination of rounding and filleting, which can be viewed as undesirable if the objective is to produce the most nearly regular set possible. We can overcome this limitation by using Mason to generate the initial condition for the flow.

Another alternative, which is also self-dual, is to change the constraints. Rather than perform the flow in $M_r\left(S\right)$, we can perform it in $M_r\left(Mason\left(S\right)\right)$. The mortar of $Mason\left(S\right)$ is a subset of the mortar of S whose concave boundary is also r-smooth. The result is not an r-tightening of S, but it is r-smooth and only differs from S in the mortar. Using $M_r\left(Mason\left(S\right)\right)$ is mostly of interest because it may make curvature flow converge faster. $M_r\left(Mason\left(S\right)\right)$ can be significantly thinner than $M_r\left(S\right)$, so that the front crosses fewer pixels during its evolution.

4.3 Medial Axis

Morphological operations can be formulated in terms of the medial axis. The medial axis $MA\left(S\right)$ is the set of all points that have more than one closest point on ∂S; note that $MA\left(S\right)$ may contain points in $\bar{S}$ as well as S. With each point q of $MA\left(S\right)$ we can associate the distance $r\left(q\right)$ from q to ∂S and the disk $C\left(q\right)$ with center q and radius $r\left(q\right)$. We can then express S as the union of $C\left(q\right)$ for all $q \in MA\left(S\right) \cap S$. We can similarly express the core as the union of $C\left(q\right)$ for all $q \in MA\left(S\right) \cap S$ such that $r\left(q\right) > r$. We can obtain the anticore from $MA\left(S\right) \cap \bar{S}$, while the mortar is the union of $C\left(q\right)$ for $r\left(q\right) \le r$. Reconstruction of a shape from a subset of the medial axis is reminiscent of the simplification algorithm used in [25].

The local feature size $lfs\left(p\right)$ is defined for $p \in \partial S$ as the distance between p and $MA\left(S\right)$. [1] Regularity can be defined in terms of local feature size: a set S is r-regular iff the minimum value of $lfs\left(p\right)$ is greater than or equal to r.

4.4 Topology Simplification

Tightening can simplify topology by merging nearby shapes and filling small holes in S and $\bar{S}$. This leads us to compare it to other topology-simplifying techniques. Many techniques seek to reduce the number of elements in a polygonal mesh; several authors have found that allowing topology changes enables more dramatic reductions in vertex count. Some approaches use topology-modifying decimation operations, such as vertex merges [9, 19]. Other approaches, however, perform topology simplification separate from decimation. They include using standard morphological operations [8, 18], volumetric lowpass filtering, [13] and explicit handle removal [24, 30]. We discussed the standard morphological operations above. Although, like tightening, low-pass filtering can round off convexities and fill in concavities, it does not necessarily increase the minimum radius

of curvature. Furthermore, it tends to erode the shape along regular portions that tightening would preserve. Finally, the explicit handle reduction techniques are not concerned with smoothness or regularity.

4.5 Fairing

Tightening can be related to surface fairing methods, which we can broadly categorize into optimization-based approaches and polygon mesh smoothing. The optimization-based approaches appear in the CAD/CAM literature on problems such as ship hull and car body design [4, 12, 17, 28]. The objective of these methods is to deform a surface composed of curved patches to minimize an energy function. A variety of functions have been proposed. Some directly penalize high curvature by, for instance, penalizing integral square curvature. Like tightening, minimizing such a function may decrease maximum curvature, although it would not guarantee a curvature bound. Other functions, such as those that penalize variation in curvature, have a more tenuous link to tightening.

There is also an extensive computer graphics literature on polygon mesh smoothing, which is often used to eliminate acquisition artifacts from laser range scanning or isosurface extraction. Mesh smoothing techniques can be variously related to robust statistics [10, 15], signal processing [26, 31], and partial differential equations [3, 6]. The goals of tightening can be distinguished from those of techniques such as bilateral filtering and anisotropic diffusion that seek to eliminate noise while preserving sharp features and consequently might not reduce maximum curvature. Our use of curve evolution differs from those of the remaining techniques in that our objective is to shrink the shape as quickly as possible while depending on the mortar to ensure that this shrinking both eliminates high curvature and preserves low curvature parts of the boundary. Other approaches, which do not employ a tolerance zone, seek to selectively affect high-curvature boundary while minimizing or correcting for the effect of evolution on the gross shape. For instance, the use of curvature flow in [6] relies on the fact that it dampens small-scale, high-curvature oscillations more quickly than it affects the slowly varying parts of the shape. The authors propose to correct for the global shrinking that occurs by rescaling the surface to preserve volume, which can displace the flatter contours that the mortar preserves. Similarly, signal processing approaches are designed to attenuate high frequencies while retaining low frequencies. They were in fact originally developed to eliminate the shrinking associated with Laplacian smoothing [26]. To the extent that frequency can be equated with curvature, tightening can be considered a filter with a precise cutoff: it perfectly preserves those parts of the curve with curvature less than r (provided they are regular,) while completely removing parts with curvature greater than r.

4.6 Relative Convex Hull

The idea of finding a set bounded by the shortest path around a core set inside an outer set appears in Sklansky's work on the minimum perimeter polygon [23]. This set was later termed the relative convex hull or geodesic convex hull. If the core and outer sets are both simple polygons with $O(n)$ sides, the relative convex hull can be computed using path planning in a simple polygon, which can be performed in $O(n)$ time [16], while if the core is a set of $O(n)$ points and the outer set is a polygon of $O(n)$ sides the relative convex hull requires $O(n \log n)$ time [27]. The link between convexity and minimum boundary is specific to two dimensions; the minimum surface enclosing a solid is not necessarily convex.

5. CONCLUSION

We have defined r-tightenings for input sets of arbitrary topology. We have shown that although an r-tightening of a set only differs from that set in the mortar, its boundary is guaranteed to be r-smooth. Our definition is independent of shape representation, and it may be possible to compute tightenings using a variety of algorithms. We have described how to compute tightenings for sets represented as binary images using constrained, level-set curvature flow.

6. ACKNOWLEDGMENTS

This research was supported by a DARPA/NSF CARGO grant number 0138420.

7. REFERENCES

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry* 22 (1999) 481-504.

[2] D. Attali. r-Regular shape reconstruction from unorganized points. *Computational Geometry* (1997) 248-253.

[3] C. Bajaj and G. Xu. Anisotropic diffusion of surfaces and functions on surfaces. *ACM Transactions on Graphics* 22, 1 (2003) 4-32.

[4] G. Celniker and D. Gossard. Deformable curve and surface finite elements for free-form shape design. *SIGGRAPH 91* (1991) 257-266.

[5] J. Cohen, A. Varshney, et. al. Simplification envelopes. *SIGGRAPH 96* (1996) 119-128.

[6] M. Desbrun, M, Meyer, et. al. Implicit fairing of irregular meshes using diffusion and curvature flow. *SIGGRAPH 99* (1999) 317-324.

[7] P. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing* 14 (1980) 227-248.

[8] J. El-Sana and A. Varshney. Topology simplification for polygonal virtual environments. *IEEE Transactions on Visualization and Computer Graphics* 4, 2 (1998) 133-144.

[9] C. Erikson and D. Manocha. GAPS: General and automatic polygonal simplification. *Proceedings of the 1999 Symposium on Interactive 3D Graphics* (1999) 79-88.

[10] S. Fleischman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *SIGGRAPH 2003* (2003) 950-953.

[11] M. Grayson, The heat equation shrinks embedded plane curves to round points. Journal of Differential Geometry 26 (1987) 285-314.

[12] S. Hahmann and S. Konz. Fairing bi-cubic B-spline surfaces using simulated annealing, in: A. Le Mehaute, C. Rabut, and L. Schumaker (Eds.), *Curves and Surfaces with Applications in CAGD*,.Vanderbilt University Press, Nasheville, TN, 1997, pp. 159-168.

[13] T. He, L. Hong, et. al. Controlled topology simplification. *IEEE Transactions on Visualization and Computer Graphics* 2, 2 (1996) 171-183.

[14] H. Heijmans. *Morphological Image Operators*. Academic Press, New York, 1994.

[15] T. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *SIGGRAPH 2003* (2003) 943-949.

[16] J. Mitchell. Shortest paths and networks, in: J. Goodman and J. O'Rourke (Eds.), *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, FL, 2004, pp. 607-641.

[17] H. Moreton and C. Sequin. Functional optimization for fair surface design. SIGGRAPH 92 (1992) 167-176.

[18] F. Nooruddin and G. Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics* 9, 2 (2003) 191-205.

[19] J. Popovic and H. Hoppe. Progressive simplicial complexes. *SIGGRAPH 97* (1997) 217-224.

[20] J. Rossignac. Blending and offsetting solid models. Ph. D. thesis, University of Rochester, NY, 1985.

[21] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[22] J. Sethian. Level Set Methods and Fast Marching Methods. Cambridge University Press, 1999.

[23] J. Sklansky, R. Chazin, and B. Hansen. Minimum-perimeter polygons of digitized silhouettes. *IEEE Transactions on Computers* 21, 3 (1972) 260-268.

[24] A. Szymczak and J. Vanderhyde. Extraction of topologically simple isosurfaces from volume datasets. *IEEE Visualization 2003* (2003) 67-74.

[25] R. Tam and W. Heidrich. Shape simplification based on the medial axis transform. *IEEE Visualization 2003* (2003) 481-488.

[26] G. Taubin. Curve and surface smoothing without shrinkage. *Fifth International Conference on Computer Vision* (1995) 852-857.

[27] G. Toussaint. Computing geodesic properties inside a simple polygon. *Revue D'Intelligence Artificielle* 3, 2 (1989) 9-42.

[28] G. Westgard and H. Nowacki. Construction of fair surfaces over irregular meshes. *ACM Symposium on Solid Modeling and Applications* (2001) 88-98.

[29] J. Williams and J. Rossignac. Mason: morphological simplification. To appear in *Graphical Models*.

[30] Z. Wood, H. Hoppe, et. al. Removing excess topology from isosurfaces. *ACM Transactions on Graphics* 23, 2 (2004) 190-208.

[31] H. Zhang and E. Fiume. Butterworth filtering and implicit fairing of irregular meshes. *11th Pacific Conference on Computer Graphics and Applications* (2003) 502-506.